

This listing of claims will replace all prior versions, and listings, of claims in the application.

Listing of Claims:

1-9. (Cancelled)

10. (Currently Amended) A method for improving the efficiency of a message processing system, comprising:

determining a workload of a message processing system by accessing performance data regarding the message processing system, and determining, using the performance data, the workload with respect to a system operating parameter;

polling for a new message at a frequency according to the workload status, wherein the frequency is inversely proportional to the workload and, if the workload is above a predetermined limit, polling only for a new non-activation message;

identifying a blocked instance being processed by the message processing system;

calculating an expected idle time for the blocked instance by:

accessing performance data for the message processing system;

determining a length of time the blocked instance has been idle[:]; and

generating the expected idle time based on the performance data and length of time the blocked instance has been idle;

dehydrating the blocked instance if the expected idle time exceeds a predetermined threshold;

updating the workload according to the dehydration of the instance; and

updating the threshold according to the workload.

11. (Cancelled)

12. (Previously Presented) The method of claim 10, wherein the accessed performance data is memory usage.

13. (Previously Presented) The method of claim 10, wherein the accessed performance data is processor power in use by the message processing system.

14-16. (Cancelled)

17. (Original) The method of claim 10, wherein the polling step is carried out at one of a first or second frequencies, wherein the first frequency is greater than the second frequency.

18. (Original) The method of claim 17, wherein the polling step further comprises polling only for a new non-activation message.

19. (Cancelled)

20. (Previously Presented) The method of claim 10, wherein the performance data is assigned according to a predetermined criterion if no performance data is accessible.

21. (Original) The method of claim 10, wherein the blocked instance is a first instance, and the performance data comprises a recorded idle time of a second instance.

22. (Cancelled)

23. (Currently Amended) A method for managing a workload of a message processing system, comprising:

determining the workload of the message processing system;

polling for a new message at a frequency, wherein the frequency is inversely proportional to the workload and, if the workload is above a predetermined limit, polling only for a new non-activation message;

identifying a blocked instance being processed by the message processing system and, if the blocked instance has no executable segments:

calculating an expected idle time for the blocked based on performance data relating to the message processing system ~~instance~~ by:

accessing performance data for the message processing system;

determining a length of time the blocked instance has been idle[:];
and
generating the expected idle time based on the performance data and
length of time the blocked instance has been idle; [[and]]
determining whether the expected idle time exceeds a predetermined threshold
and, if so,
dehydrating the blocked instance;
updating the workload according to the dehydration; and
updating the performance data according to the polling of the new message.

24. (Cancelled)

25. (Currently Amended) A computer-readable storage medium having computer-readable instructions for performing a method for improving the efficiency of a message processing system, the method comprising:

determining a workload of a message processing system by accessing performance data regarding the message processing system, and determining, using the performance data, the workload with respect to a system operating parameter;

polling for a new message at a frequency according to the workload status, wherein the frequency is inversely proportional to the workload and, if the workload is above a predetermined limit, polling only for a new non-activation message;

identifying a blocked instance being processed by the message processing system;

calculating an expected idle time for the blocked instance by:

accessing performance data for the message processing system;

determining a length of time the blocked instance has been idle[:]; and

generating the expected idle time based on the performance data and length of time the blocked instance has been idle;

dehydrating the blocked instance if the expected idle time exceeds a predetermined threshold;

updating the workload according to the dehydration of the instance; and

updating the threshold according to the workload.

26. (Cancelled)

27. (Previously Presented) The computer-readable medium of claim 25, wherein the accessed performance data is memory usage.

28. (Previously Presented) The computer-readable medium of claim 25, wherein the accessed performance data is processor power in use by the message processing system.

29-31. (Cancelled)

32. (Previously Presented) The computer-readable medium of claim 25, wherein the polling step is carried out at one of a first or second frequencies, wherein the first frequency is greater than the second frequency.

33. (Previously Presented) The computer-readable medium of claim 32, wherein the polling step further comprises polling only for a new non-activation message.

34. (Cancelled)

35. (Previously Presented) The computer-readable medium of claim 25, wherein the performance data is assigned according to a predetermined criterion if no performance data is accessible.

36. (Previously Presented) The computer-readable medium of claim 25, wherein the blocked instance is a first instance, and the performance data comprises a recorded idle time of a second instance.

37. (Cancelled)

38. (Currently Amended) A computer-readable storage medium having computer-executable instructions for performing a method for managing a workload of a message processing system, the method comprising:

determining the workload of the message processing system;

polling for a new message at a frequency, wherein the frequency is inversely proportional to the workload and, if the workload is above a predetermined limit, polling only for a new non-activation message;

identifying a blocked instance being processed by the message processing system and, if the blocked instance has no executable segments:

calculating an expected idle time for the blocked instance based on performance data relating to the message processing system ~~instance~~ by:

accessing performance data for the message processing system;

determining a length of time the blocked instance has been idle[[:]]; and

and

generating the expected idle time based on the performance data and length of time the blocked instance has been idle; [[and]]

determining whether the expected idle time exceeds a predetermined threshold and, if so,

dehydrating the blocked instance;

updating the workload according to the dehydration; and

updating the performance data according to the polling of the new message.

39. (Cancelled)